

PERBANDINGAN KINERJA METODE DETEKSI TEPI PADA CITRA

Kornelis Letelay

Jurusan Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana

Email : kletelay@gmail.com

INTISARI

Berdasarkan hasil analisis bahwa deteksi tepi paling baik dihasilkan dari pengguna metode *canny*. Deteksi tepi dengan menggunakan metode *canny* adalah deteksi tepi terbaik dikarenakan morfologi garis yang dihasilkan oleh deteksi tepi ini lebih jelas terlihat pada garis tepi gambar, baik pada bagian dalam maupun tepi gambar terlihat tebal, garis vertikal maupun horizontal pada bagian depan obyek sangat jelas jika dibandingkan dengan dua metode di atas.

Kata Kunci : deteksi tepi, *image*, metode *canny*

ABSTRACT

The results of the above analysis it is concluded that edge detection is best resulted from the user of the *canny* method. Edge detection using the *canny* method is the best edge detection because the line morphology generated by edge detection is better visible on the border of the image both on the inside and the edges of the image appear thick, vertical or horizontal lines on the front of the house is very clear when compared with the two methods above.

Keywords: *Edge detection*, *image*, *canny method*

I. PENDAHULUAN

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Pengolahan citra telah menggunakan sistem komputer yang diaplikasikan pada sejumlah bidang, seperti pada bidang kedokteran, biologi, hukum, dan keamanan.

Teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, tetapi citra keluarannya harus mempunyai kualitas lebih baik daripada citra masukan. Operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis yaitu perbaikan kualitas citra, pemugaran citra, segmentasi citra, analisis citra, dan rekonstruksi citra.

Peningkatan kualitas citra bertujuan menghasilkan citra dengan kualitas yang lebih baik dibandingkan dengan citra semula. Analisis citra bertujuan mengidentifikasi parameter yang diasosiasikan dengan ciri dari objek dalam citra, untuk selanjutnya parameter tersebut digunakan dalam menginterpretasi citra. Analisis citra pada dasarnya terdiri dari tiga tahapan, yaitu ekstraksi ciri, segmentasi, dan klasifikasi. Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi di dalam citra. Ada beberapa metode deteksi tepi. Penggunaan metode deteksi tepi yang tidak tepat, akan menghasilkan pendeteksian yang gagal. Pendeteksian tepi merupakan tahapan untuk melingkupi informasi di dalam citra. Tepi mencirikan batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra (Munir, 2004).

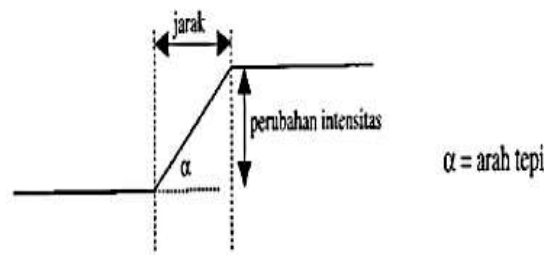
Dalam penelitian ini citra yang akan digunakan sebagai uji coba adalah suatu citra diam berwarna dengan format JPEG/JPG. Tujuan dari penelitian ini adalah membandingkan beberapa metode deteksi tepi pada citra bangunan rumah.

II. MATERI DAN METODE

2.1 Konsep Deteksi Tepi

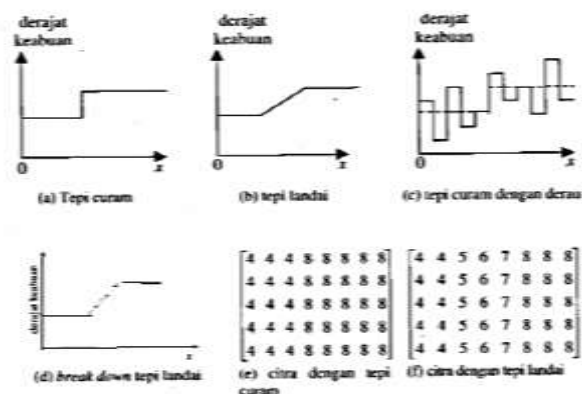
Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang memperlihatkan rincian pada gambar. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung pada perubahan intensitas.

Deteksi tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan operasi deteksi tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra (Munir, 2004).



Gambar 1. Model Tipe satu-Matra

Ada tiga macam tepi yang terdapat di dalam citra digital. Ketiganya adalah tepi curam, tepi landai, dan tepi yang mengandung derau.



Gambar 2. Jenis-jenis Tepi (Munir, 2004)

2.2 Teknik Deteksi Tepi

Terdapat beberapa teknik yang digunakan untuk mendeteksi tepi, antara lain (Munir, 2004):

1. Operator gradien pertama, contoh beberapa gradien pertama yang dapat digunakan untuk mendeteksi tepi di dalam citra, yaitu operator gradien selisih-terpusat, operator *Sobel*, operator *Prewitt*, operator *Roberts*, operator *Canny*.
2. Operator turunan kedua, disebut juga operator *Laplace*. Operator *Laplace* mendeteksi lokasi tepi khususnya pada citra tepi yang curam. Pada tepi yang curam, turunan keduanya mempunyai persilangan nol, yaitu titik di mana terdapat pergantian tanda nilai turunan kedua, sedangkan pada tepi yang landai tidak terdapat persilangan nol. Contohnya adalah operator *Laplacian Gaussian*, operator *Gaussian*.

1. Operator kompas, digunakan untuk mendeteksi semua tepi dari berbagai arah di dalam citra. Operator kompas yang dipakai untuk deteksi tepi menampilkan tepi dari 8 macam arah mata angin yaitu Utara, Timur Laut, Timur, Tenggara, Selatan, Barat, Barat Daya, dan Barat Laut. Deteksi tepi dilakukan dengan mengkonvolusikan citra dengan berbagai *mask* kompas, lalu dicari nilai kekuatan tepi (*magnitude*) yang terbesar dan arahnya. Operator kompas yang dipakai untuk deteksi tepi menampilkan tepi dari 8 macam arah mata angin, yaitu Utara, Timur Laut, Timur, Tenggara, Selatan, Barat, Barat Daya, dan Barat Laut.

Selain operator gradien yang sudah disebutkan, masih ada beberapa operator gradien yang lain yang dapat digunakan untuk mendeteksi tepi di dalam citra, yaitu sobel, prewitt, dan Canny.

(a) Operator *Sobel* Tinjau pengaturan *pixel* di sekitar *pixel* (x,y) :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator *Sobel* adalah magnitudo dari gradien yang dihitung dengan

$$M = \sqrt{s_x^2 + s_y^2}$$

yang dalam hal ini, turunan parsial dihitung dengan :

$$s_x = (a_7 + ca_3 + a_4) - (a_0 + ca_1 + a_6)$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

dengan konstanta $c = 2$. Dalam bentuk *mask*, s_x dan s_y dapat dinyatakan sebagai

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Arah tepi dihitung dengan persamaan :

$$\alpha(x,y) = \tan^{-1} \left(\frac{s_y}{s_x} \right)$$

Contoh berikut ini memperlihatkan deteksi tepi dengan operator *Sobel*. Konvolusi pertama dilakukan terhadap *pixel* yang bernilai 1 (di titik pusat *mask*):

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

(i) Citra Semula

$$\begin{bmatrix} * & * & * & * & * \\ * & 18 & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

(ii) hasil konvolusi

Nilai 18 pada citra hasil konvolusi diperoleh dengan perhitungan berikut :

$$s_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$s_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

pada contoh ini nilai:

$$M = \sqrt{s_x^2 + s_y^2}$$

dihampiri dengan menghitung :

$$M \cong |S_x| + |S_y|$$

Di bawah ini contoh lain deteksi tepi dengan operator *Sobel*, di mana hasil konvolusi diibandingkan dengan $T = 12$.

Citra:

0	0	0	0	0	0	2	0	3	3
0	0	0	3	0	0	0	2	4	3
0	0	2	0	2	4	3	3	2	3
0	0	1	3	3	4	3	3	3	3
0	1	0	4	3	3	2	4	3	2
0	0	1	2	3	3	4	4	4	3

| gradien - x | + | gradien - y |:

*	*	*	*	*	*	*	*	*	*
*	4	6	4	10	14	12	14	4	*
*	6	8	10	20	16	12	6	0	*
*	4	10	14	10	2	4	2	4	*
*	2	12	12	2	2	4	6	8	*
*	*	*	*	*	*	*	*	*	*

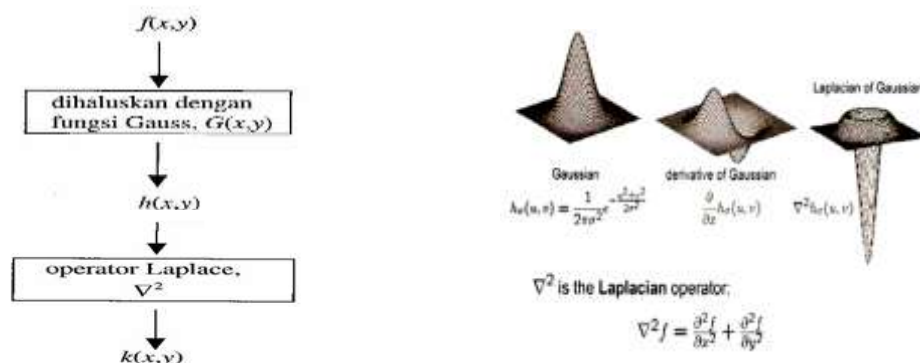
Hasil pengambangan dengan $T = 12$:

*	*	*	*	*	*	*	*	*	*
*	4	6	4	10	14	12	14	4	*
*	6	8	10	20	16	12	6	0	*
*	4	10	14	10	2	4	2	4	*
*	2	12	12	2	2	4	6	8	*
*	*	*	*	*	*	*	*	*	*

- (b) Operator *Prewitt* Persamaan gradien pada operator *Prewitt* sama seperti operator *Sobel*, tetapi menggunakan nilai $c = 1$:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

- (c) Operator *Canny* Berdasarkan wikipedia, operator *Canny* merupakan deteksi tepi yang optimal. Operator *Canny* menggunakan *Gaussian Derivative Kernel* untuk menyaring kegaduhan dari citra awal untuk mendapatkan hasil deteksi tepi yang halus.



Gambar 3. *Gaussian Derivative Kernel*

III. HASIL DAN PEMBAHASAN

1. Algoritma *Grayscale*

Sebelum dilakukan proses deteksi tepi terlebih dahulu dilakukan proses *Gray* pada citra yang akan dianalisis, berikut adalah algoritma dari proses *grayscale*.

```
//proses grayscale Integer
i,j,nilai; i=0;j=0;
r,g,b=array()
for i ← 0 to width do
    for j ← 0 to height do
        r[j][i]←(rgb>>16) & 0xFF
        g[j][i]←(rgb>>8) & 0xFF
        b[j][i]←rgb
    endfor
endfor
Listing program metode edge detection :
Dim w(700, 300) As Integer
n1 = 0
For i = 1 To gbAwal.ScaleWidth Step 15
    n1 = n1 + 1
    n2 = 0
    For j = 1 To gbAwal.ScaleHeight Step 15
        warna = gbAwal.Point(i, j)
        r = warna And RGB(255, 0, 0)
        g = Int((warna And RGB(0, 255, 0)) / 256)
        b = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)
        wx = Int((r + g + b) / 3)
        gbGray.PSet (i, j), RGB(wx, wx, wx)
        n2 = n2 + 1
        w(n1, n2) = wx
    Next j
Next i
For i = 1 To n1
    For j = 1 To n2
        If i = 1 Then wx1 = w(i, j) Else wx1 = w(i, j) - w(i - 1, j)
        If j = 1 Then wx2 = w(i, j) Else wx2 = w(i, j) - w(i, j - 1)
        wx = Abs(wx1) + Abs(wx2)
        If wx > 255 Then wx = 255
        Metode ((i - 1) * 15 + 1, (j - 1) * 15 + 1), RGB(wx, wx, wx)
    Next j
Next i
End Sub
```

Gambar di bawah ini merupakan gambar asli dan hasil proses *grayscale*.



Gambar 4. Citra Asli

Gambar 5. Tampilan *Grayscale*

2. *Edge Detection*

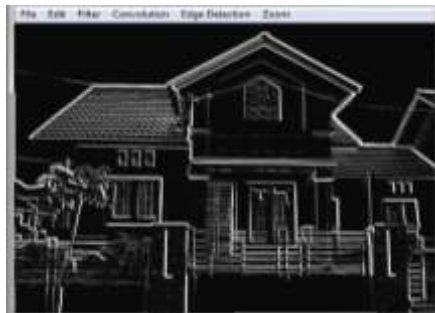
Proses berikutnya adalah melakukan deteksi tepi dengan metode yang diusulkan dengan matrik yang berbeda.

a. Metode *Prewitt*

Matrik yang digunakan dalam proses deteksi tepi dengan metode *Prewitt* adalah :

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Di bawah ini merupakan hasil dari metode *Prewitt*

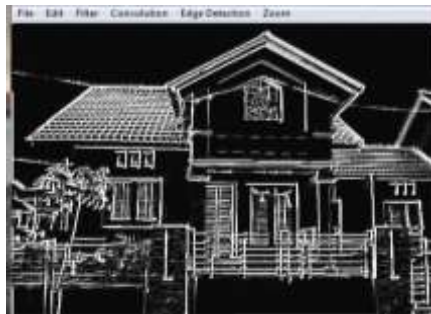
Gambar 6. Tampilan metode *Prewitt*

Analisis citra: pada tampilan dengan metode *Prewitt* terlihat tepi citra tebal dan jelas dan garis vertical dan horizontal pada bagian dalam citra juga cukup jelas, walaupun masih terdapat garis putus-putus.

b. Metode *Sobel*

Matrik yang digunakan dalam proses deteksi tepi dengan metode *Sobel* adalah:

Di bawah ini merupakan hasil dari metode *Sobel*.

Gambar 7. Tampilan metode *Sobel*

Analisis citra: pada tampilan dengan menggunakan metode *Sobel* lebih tajam setiap garis pada setiap sisi rumah dibandingkan menggunakan metode Prewitt, terlihat perbedaan pada garis vertical dan horizontal pada setiap sudut rumah baik di atas dan juga tepi pada gambar.

c. Metode *Canny*

Matrik yang digunakan dalam proses deteksi tepi dengan metode *Canny* adalah :

Gambar 8. Tampilan metode *canny*

Analisis citra: dengan menggunakan metode *canny* terlihat garis tepi pada gambar baik pada bagian dalam maupun tepi gambar terlihat tebal, garis vertical maupun horizontal pada bagian depan rumah sangat jelas jika dibandingkan dengan dua metode di atas.

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil analisis diatas maka disimpulkan bahwa deteksi tepi paling baik dihasilkan dari pengguna metode *canny*. Deteksi tepi dengan menggunakan metode *canny* adalah deteksi tepi terbaik dikarenakan morfologi garis yang dihasilkan oleh deteksi tepi ini lebih baik terlihat pada garis tepi gambar, baik pada bagian dalam maupun tepi gambar terlihat tebal, garis vertical maupun horizontal pada bagian depan obyek sangat jelas jika dibandingkan dengan dua metode di atas.

4.2. Saran

Pada penelitian selanjutnya perlu di tambahkan metode deteksi tepi yang lain seperti metode Robert, untuk di jadikan perbandingan pada kasus yang sama atau kasus yang berbeda, sehingga dapat membedakan hasil kerja metode deteksi tersebut dengan metode deteksi yang digunakan.

DAFTAR PUSTAKA

- A. P. Asy'ari, *Deteksi Tepi Khat arab menggunakan Operator Sobel dan Canny*, vol. 1, 2015.
- Balza Achmad & Kartika Firdausy 2005. *Teknik Pengolahan Citra Digital*
- Daniel. Christopher, *Edge Detection dengan Algoritma Canny*, pp. 198- 203.
- Rinaldi Munir. 2004. *Pengolahan Citra Digital*. Bandung: Informatika
- S. Elisabeth Denis Setiani, "New Edge Detection Method Using Elisabeth Method: Case Study Javanese Batik," *Jurnal Buana Informatika*, vol. I, pp. 47- 56, 2010.
- Weibin. Rong, Zhianjing. Li dan Sun. Lining, *An Improved Canny Edge Detection Algorithm*, pp. 577-582, 2014.
- Yodha.,. Johanes. Widagho. Kurniawan dan A. Wahid, "Perbandingan Penggunaan Deteksi Tepi dengan metode Laplacian, Sobel, prewitt dan Canny pada Pengenalan Pola," *Techno.COM*, vol. 13, pp. 189-197, 2014.